Hardening DGA Classifiers Utilizing IVAP

Charles Grumer*, Jonathan Peck†§, Femi Olumofin‡, Anderson Nascimento*, Martine De Cock*†
* *School of Engineering and Technology, University of Washington, Tacoma, USA*
Email: {cgrumer, andclay, mdecock}@uw.edu
† *Dept. of Applied Math., Computer Science, and Statistics, Ghent University, Ghent, Belgium*
Email: {jonathan.peck,martine.decock}@ugent.be
‡ *Infoblox, Santa Clara, USA*
Email: folumofin@infoblox.com
§ *Data Mining and Modeling for Biomedicine, VIB Inflammation Research Center, Ghent, Belgium*

***Abstract*—Domain Generation Algorithms (DGAs) are used by malware to generate a deterministic set of domains, usually by utilizing a pseudo-random seed. A malicious botmaster can establish connections between their command-and-control center (C&C) and any malware-infected machines by registering domains that will be DGA-generated given a specific seed, rendering traditional domain blacklisting ineffective. Given the nature of this threat, the real-time detection of DGA domains based on incoming DNS traffic is highly important. The use of neural network machine learning (ML) models for this task has been well-studied, but there is still substantial room for improvement. In this paper, we propose to use Inductive Venn–Abers predictors (IVAPs) to calibrate the output of existing ML models for DGA classification. The IVAP is a computationally efficient procedure which consistently improves the predictive accuracy of classifiers at the expense of not offering predictions for a small subset of inputs and consuming an additional amount of training data.**

***Keywords*-domain generation algorithms, Inductive Venn-Abers predictors, neural network**

I. Introduction

Botmasters commonly use domain generation algorithms (DGAs) to form connections between their command-and-control centers (C&C) and malware-infected machines [1]. The ability to dynamically generate new domain names prevents ordinary blacklisting from effectively blocking access between the botmaster and infected machines, highlighting the need for real-time classifiers that can accurately detect DGAs in DNS traffic (see e.g. [2], [3], [4], [5]). The effectiveness of various DGA classifiers on such data was explored in [6]. Here, we improve these results using Inductive Venn-Abers Predictors (IVAPs), which were introduced by [7]. This methodology improves binary classification results at the expense of not offering predictions for a small number of inputs and decreasing the training data volume to form an additional calibration data set.

Specifically, the IVAP algorithm uses an additional held-out *calibration set* in order to gauge the reliability of the predictions made by an existing model. We exploit this property to detect when the model is too unreliable on a given sample. By rejecting predictions if the uncertainty is too high, we are able to significantly increase true positive rates while maintaining the same low false positive rate. In deployed DGA detection systems, a low false positive rate is very important, because blocking legitimate traffic is highly undesirable. In practice, predictions that were rejected because they were found too unreliable, may be fed into a more complex model or deferred to human experts.

II. Raw Data

This paper uses raw data collected from three sources: Alexa, Qname, and Bambenek. The data sets utilized are the same as those used in [8].

Alexa[1] offers a list of the top one million domains based on their popularity in terms of number of page views and number of unique visitors. It only retains the websites' second level domain names (SLDs), aggregating across any subdomains. For example, according to Alexa, the three highest ranked domain names in terms of popularity on 2019-11-19 are *google.com*, *youtube.com*, and *tmall.com*. This Alexa top 1 million list serves as a relatively reliable source for benign, non-DGA generated domains but is not necessarily an accurate representation of benign web traffic as whole.

Qname contains domain names originating from a real-time stream of passive DNS data that consists of roughly 10-12 billion DNS queries per day collected from subscribers including ISPs (Internet Service Providers), schools, and businesses. We retained 1 million domain names that match three criteria: (1) the domain is at least 30 days old, (2) the domain has been resolved at least twice, and (3) queries to the domain have never resulted in an NXDomain response. This data set is intended to serve as a source of benign ground truth that is more reflective of real web traffic than Alexa.

Bambenek[2] offers a daily feed of domains generated by reverse engineering known families of malware. One million different DGA domains were collected over the course of three days to construct a malicious data set [8].

[1]https://www.alexa.com/topsites
[2]https://osint.bambenekconsulting.com/feeds/

III. Deep Learning Architectures

We utilize three neural network architectures that have been well-reviewed in past DGA classification literature:

- *LSTM.MI* is a unidirectional LSTM recurrent neural network architecture original proposed in [5] that has been shown to be highly effective at DGA detection [6], [8].
- *Invincea* is a neural network architecture created in [3] that features parallel convolution layers.
- *MIT* is a hybrid CNN/RNN neural network architecture introduced in [9] and shown to be effective for inline DGA detection in [10]. It features stacked CNN layers followed by a unidirectional LSTM layer.

IV. Method

We constructed two data sets, AlexaBamb and QnameBamb, from the raw data from Section II. Both data sets utilize 1 million Bambenek domains as their source of malicious data, and feature 1 million benign domains from Alexa and Qname respectively. The data sets were split into four parts: 64% training, 4% validation, 16% calibration, and 16% testing. We trained models with the architectures from Section III for a maximum of 100 epochs with early stopping on the validation loss set at 10 epochs. The final results, as reported in section V, were calculated using the 16% testing data.

The IVAP method works by taking an existing ML model F as well as a held-out calibration set. It uses these data to calibrate the predictions of F so that, at inference time, each sample x can be associated with two probabilities p_0 and p_1, satisfying the following property [7]:

$$p_0 \leq \Pr[Y = 1 \mid X = x] \leq p_1.$$

That is, they form bounds on the true probability that the label is 1 given the input. A natural measure of uncertainty of the IVAP is the width of this interval, $p_1 - p_0$. Our method rejects predictions where the uncertainty exceeds a specified threshold β, which is tuned on the separate validation set. More concretely, we use the underlying model F to predict labels and use the IVAP to quantify the uncertainty of those predictions in the form of the probability interval $[p_0, p_1]$. If $p_1 - p_0 \leq \beta$, we simply return the prediction of F. Otherwise, we reject the prediction, signaling that the model F is too unreliable on the given sample to be of any use.

As noted by [7], the computational overhead of this approach is $\mathcal{O}(m \log m)$ where m is the size of the calibration data set. Furthermore, as the size of the calibration data set increases, the difference $p_1 - p_0$ tends to decrease, making it more likely that IVAP rejections are correct (in the sense that the underlying model is probably wrong when the uncertainty threshold is exceeded).

V. Results

The results are shown in Table I. For both the QnameBamb and AlexaBamb data sets, we report results for each of the models (LSTM.MI, Invincea, and MIT) with and without the IVAP. The metrics of interest for the baseline models are the true positive rate (TPR), false positive rate (FPR) and accuracy (ACC). To enable a fair comparison with the IVAP, we compute these metrics only for the subset of samples that were not rejected by the IVAP-augmented model. For the IVAP, additional metrics include the false rejection rate (FRR), true rejection rate (TRR) and overall rejection rate (REJ). Here, a rejection is *true* when the underlying model prediction was indeed wrong and *false* otherwise. The rejection rate is simply the fraction of samples for which predictions were rejected.

The uncertainty threshold β was tuned on the separate validation set in order to maximize the difference $\mathrm{TRR} - \mathrm{FRR}$ similarly to Youden's index [11]. Note that this is just one of many possible ways one could tune β. In some applications, one might prefer the lowest possible FPR, for example. Moreover, since β is just a single scalar hyperparameter, it is also possible to specify it manually without resorting to potentially expensive hyperparameter optimization methods on held-out data.

The use of IVAPs results in consistently better predictive scores at the cost of a small rejection rate and the need for additional data sets, which is in line with [7]. Most notably, the TPR always increases across data sets and models after the IVAP process is applied while the false positive rate remains stable. This suggests that the IVAP process can be used to improve the performance of existing DGA classifiers. One exception to this is the MIT model on the QnameBamb data set: here, all predictions were rejected because the probability interval was always larger than the tuned threshold would allow. We speculate that this is due to the QnameBamb data set being more difficult to fit properly, as evidenced by the fact that the classifiers perform consistently better on AlexaBamb than QnameBamb. This means that models trained on QnameBamb might require a larger calibration set to obtain good results with the IVAP. We leave a more in-depth exploration of the failure modes of the IVAP to future work.

VI. Conclusion

We have proposed a computationally efficient procedure for hedging the predictions of DGA classifiers. Our method allows us to detect when these models are too unreliable on a given sample. By rejecting predictions if the uncertainty is too high, we achieve consistently higher predictive performance across different models and data sets. The price to pay for this increased performance is a smaller amount of data available for training (as the method needs to be calibrated on a separate held-out data set) as well as a small fraction of samples for which we cannot give any prediction. Such rejected predictions need to be deferred to more complex models or to human experts.

Table I
COMPARISONS OF MODEL PERFORMANCE WITH AND WITHOUT IVAP

Data Set	Classifier	TPR	FPR	ACC	FRR	TRR	REJ
QnameBamb	LSTM.MI	0.917	0.001	0.993	-	-	-
	LSTM.MI + IVAP	0.993	0.001	0.996	0.064	0.922	0.100
	Invincea	0.856	0.001	0.991	-	-	-
	Invincea + IVAP	0.986	0.001	0.993	0.126	0.916	0.183
	MIT	0.920	0.001	0.993	-	-	-
	MIT + IVAP	-	-	-	-	-	1.000
AlexaBamb	LSTM.MI	0.964	0.001	0.992	-	-	-
	LSTM.MI + IVAP	0.999	0.001	0.999	0.066	0.950	0.082
	Invincea	0.966	0.001	0.991	-	-	-
	Invincea + IVAP	0.999	<0.001	0.999	0.108	0.969	0.123
	MIT	0.909	0.001	0.992	-	-	-
	MIT + IVAP	0.973	<0.001	0.988	0.286	0.493	0.290

FPR=False Positive Rate, ACC=Accuracy, TPR=True Positive Rate
TRR=True Rejection Rate, FRR=False Rejection Rate, REJ=Rejection Rate

While these results certainly highlight IVAP's potential for increasing the statistical measurements of DGA identification models, the IVAP can sometimes fail to calibrate properly. This is an issue which warrants further study, since it otherwise yields a useless model. Initial areas for further research include comparing various model architectures, how they affect reported loss, and how these two elements combine to affect calibration. Data noise is also of interest, as imprecisely labeled data (such as the Qname data) may also affect the accuracy of the calibration step. Establishing a relationship between the size of the calibration set and the quality of the IVAP output similar to traditional generalization bounds [12] is also an interesting avenue for future work.

REFERENCES

[1] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla, "A comprehensive measurement study of domain generating malware," in *25th USENIX Security Symposium*, 2016, pp. 263–278.

[2] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Predicting domain generation algorithms with long short-term memory networks," *preprint arXiv:1611.00791*, 2016.

[3] J. Saxe and K. Berlin, "eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys," *arXiv preprint arXiv:1702.08568*, 2017.

[4] S. Schüppen, D. Teubert, P. Herrmann, and U. Meyer, "FANCI: Feature-based automated NXDomain classification and intelligence," in *USENIX Security Symposium*, 2018, pp. 1165–1181.

[5] D. Tran, H. Mac, V. Tong, H. A. Tran, and L. G. Nguyen, "A LSTM based framework for handling multiclass imbalance in DGA botnet detection," *Neurocomputing*, vol. 275, pp. 2401–2413, 2018.

[6] R. Sivaguru, C. Choudhary, B. Yu, V. Tymchenko, A. Nascimento, and M. De Cock, "An evaluation of DGA classifiers," in *IEEE BigData*, 2018, pp. 5058–5067.

[7] V. Vovk, I. Petej, and V. Fedorova, "Large-scale probabilistic predictors with and without guarantees of validity," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 892–900.

[8] J. Peck, C. Nie, R. Sivaguru, C. Grumer, F. G. Olumofin, B. Yu, A. C. A. Nascimento, and M. De Cock, "CharBot: A simple and effective method for evading DGA classifiers," *IEEE Access*, vol. 7, pp. 91 759–91 771, 2019.

[9] S. Vosoughi, P. Vijayaraghavan, and D. Roy, "Tweet2Vec: Learning tweet embeddings using character-level CNN-LSTM encoder-decoder," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2016, pp. 1041–1044.

[10] B. Yu, J. Pan, J. Hu, A. Nascimento, and M. De Cock, "Character level based detection of DGA domain names," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 4168–4175.

[11] R. Fluss, D. Faraggi, and B. Reiser, "Estimation of the Youden Index and its associated cutoff point," *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, vol. 47, no. 4, pp. 458–472, 2005.

[12] V. Vapnik, *Statistical learning theory*. Wiley New York, 1998.